IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A program storage device, readable by a machine, tangibly embodying programming instructions to perform method steps for constructing a set of types occurring within a method ~~call graph as a representation of a program~~, the programming instructions comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and ~~one or more~~at least two methods M_M;

wherein each method M_1 in M_M has a single body B;

~~wherein for each method M_2 in M_M, the call graph representation includes a corresponding node;~~

~~wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;~~

~~identifying~~constructing for each method M in M_M, a set of ~~zero or more~~ types S_M of objects which occur therein;

~~identifying~~constructing for each field F in F_F, a set of ~~zero or more~~ types S_F of objects stored therein;

identifying one or more allocation sites inside the body B of each of method M;

determining a set of ~~directly called~~ methods M' that are directly called within~~inside~~ the body B of each method M and propagating types from the set of types $S_{M'}$ to the set of types S_M and from of the set of types S_M to the set of types $S_{M'}$; ~~and~~

determining a set of ~~virtually called~~ methods M'' that are virtually called within~~inside~~ the body B of each method M and propagating types from the set of types $S_{M''}$ to the set of types S_M and from the set of types S_M to the set of types $S_{M''}$;

determining a set of fields F that are

read in the body B of each method M, and propagating types from the set

of types S_F to the set of types S_M; and

written in the body B of each method M, and propagating types from the set of types S_M to the set of types S_F.

2. (Currently Amended) The program storage device according to claim 1, further comprising the programming instructions of:

determining the set of all types T that are allocated in the body of method M, and adding each element of the set of all types T to the set of types S_M ~~for each allocation of type T that occurs in the method M~~.

3. (Currently Amended) The program storage device according to claim 2, further comprising the programming instructions of:

for each direct call to a method~~s~~ M' in the body B of the method M performing the steps of:

adding any type that occurs in the set of types S_M and that is a subtype of the type of a parameter of the method~~s~~ M' to the set of types $S_{M'}$; and

adding any type that occurs in the set of types $S_{M'}$ and that is a subtype of a return type of the method~~s~~ M' to the set of types S_M.

4. (Currently Amended) The program storage device according to claim 3, further comprising the programming instructions of:

for each virtual call to the method~~s~~ M' in the body B of the method M:

using the set of types S_M, determine each of the methods M" that may be reached by a dynamic dispatch:

adding any type that occurs in the set of types S_M and that is a subtype of the type of a parameter of the method~~s~~ M" to ~~a~~the set of types $S_{M"}$;

adding any type that occurs in the set of types $S_{M"}$ and that is a subtype of the return type of the method~~s~~ M" to the set of types S_M.

5. (Currently Amended) The program storage device according to claim 4, further the programming instructions of:

for each field F read by the method M, adding any type that occurs in the set of types S_F to the set of types S_M ; and

for each field F with the set of all types T written by the method M, adding any type that occurs in the set of types S_M and that is a subtype of the set of all types T to the set of types S_F.

6. (Currently Amended) The program storage device according to claim 19, further comprising the programming instructions of:

using the call graph ~~computed above~~, as previously constructed, in a compiler as a basis for performing optimizations such as inlining.

7. (Currently Amended) The program storage device according to claim 19, further comprising the programming instructions of:

using the call graph ~~computed above~~, as previously constructed, in a reporting tool to report call graph information to a user.

8. (Currently Amended) A computer program product ~~program storage device, readable by a machine, tangibly embodying instructions to perform method steps~~ for constructing a set of types occurring within a method ~~call graph as a representation of a program~~, the ~~method~~the computer program product comprising:

a storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and ~~one or more~~at least two methods M_M;

wherein each method M_1 in M_M has a single body B;

wherein for each method M_2 in M_M, the call graph representation includes a

corresponding node;

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;

~~identifying~~constructing for each method M in M_M, a set of ~~zero-or-more~~ types S_M of objects which occur therein;

~~identifying~~constructing for each field F in F_F, a set of ~~zero-or-more~~ types S_F of objects stored therein;

identifying one or more allocation sites inside the body B of each of method M;

determining a set of ~~directly-called~~ methods M' that are directly called within~~inside~~ the body B of each method M and propagating types from the set of types $S_{M'}$ to the set of types S_M and from of the set of types S_M to the set of types $S_{M'}$; ~~and~~

determining a set of ~~virtually-called~~ methods M" that are virtually called within~~inside~~ the body B of each method M and propagating types from the set of types $S_{M''}$ to the set of types S_M and from of the set of types S_M to the set of types $S_{M''}$;

determining a set of fields F that are

read in the body B of each method M, and propagating types from the set of types S_F to the set of types S_M; and

written in the body B of each method M, and propagating types from the set of types S_M to the set of types S_F.

9. (Currently Amended) The computer program ~~storage-device~~product according to claim 8, further comprising:

determining the set of all types T that are allocated in the body of method M, and adding each element of the set of all types T to the set of types S_M ~~for-each-allocation-of-type-T-that-occurs-in-the-method-M~~.

10. (Currently Amended) A method for constructing a set of types occurring within a method~~call graph as a representation of a program~~, the method comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and ~~one or more~~at least two methods M_M;

wherein each method M_1 in M_M has a single body B;

~~wherein for each method M_2 in M_M, the call graph representation includes a corresponding node;~~

~~wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;~~

~~identifying~~constructing for each method M in M_M, a set of ~~zero or more~~ types S_M of objects which occur therein;

~~identifying~~constructing for each field F in F_F, a set of ~~zero or more~~ types S_F of objects stored therein;

identifying one or more allocation sites inside the body B of each of method M;

determining a set of ~~directly called~~ methods M' that are directly called within~~inside~~ the body B of each method M and propagating types from a set of types $S_{M'}$ to the set of types S_M and from of the set of types S_M to the set of types $S_{M'}$; ~~and~~

determining a set of ~~virtually called~~ methods M'' that are virtually called within~~inside~~ the body B of each method M and propagating types from a set of types $S_{M''}$ to the set of types S_M and from of the set of types S_M to the set of types $S_{M''}$;

determining a set of fields F that are

read in the body B of each method M, and propagating types from the set of types S_F to the set of types S_M; and

written in the body B of each method M, and propagating types from the set of types S_M to the set of types S_F.

11. (Currently Amended) The method according to claim 10, further comprising:

determining the set of all types T that are allocated in the body of method M, and adding each element of the set of all types T to the set of types S_M ~~for each allocation of~~

~~type T that occurs in the method M~~.

12. (Currently Amended) The method according to claim 11, further comprising:

for each direct call to ~~the~~ a methods M' in the body B of the method M performing the steps of:

adding any type that occurs in the set of types S_M and that is a subtype of a type of a parameter of the methods M' to the set of types $S_{M'}$; and

adding any type that occurs in the set of types $S_{M'}$ and that is a subtype of a return type of the methods M' to the set of types S_M.

13. (Currently Amended) The method according to claim 12, further comprising:

for each virtual call to the methods M' in the body B of the method M:

using the set of types S_M, determine each of the methods M" that may be reached by a dynamic dispatch:

adding any type that occurs in the set of types S_M and that is a subtype of a type of a parameter of the methods M" to athe set of types $S_{M"}$;

adding any type that occurs in the set of types $S_{M"}$ and that is a subtype of the return type of the methods M" to the set of types S_M.

14. (Currently Amended) The method according to claim 13, further comprising:

for each field F read by the method M, add any type that occurs in the set of types S_F to the set of types S_M ; and

for each field F with the set of all types T written by the method M, add any type that occurs in the set of types S_M and that is a subtype of the with set of all types T to the set of types S_F.

15. (Currently Amended) The method according to claim 10, further comprising the step of:

using the call graph ~~computed above~~, as previously constructed, in a compiler as a basis for performing optimizations such as inlining.

16. (Currently Amended) The method according to claim ~~1~~20, further comprising the step of:

using the call graph ~~computed above~~, as previously constructed, in a reporting tool to report call graph information to a user.

17. (Cancelled)

18. (Cancelled)

19. (New) The program storage device according to claim 1, further comprising:

constructing a call graph representation;

wherein for each method M_2 in M_M, the call graph representation includes a corresponding node; and

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes.

20. (New) The method according to claim 10, further comprising:

constructing a call graph representation;

wherein for each method M_2 in M_M, the call graph representation includes a corresponding node; and

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes.